

12

REMARKS

Applicants appreciate the thoroughness with which the Examiner has examined the above-identified application. Reconsideration is requested in view of the amendments above and the remarks below.

For purposes of appeal, no claims have been amended.

No new matter has been added.

Section 102 Issues

The Examiner has rejected claims 1-8, 12-13, 20, and 22-26 under 35 U.S.C. 102(b) as being anticipated by Cole et al. (U.S. Patent No. 5,752,042). Applicants disagree and continue to traverse this rejection.

Applicant submits that the present invention is not anticipated by Cole et al. Anticipation is but the ultimate or epitome of obviousness. To constitute anticipation, all material elements of a claim must be found in one prior art source. In re Marshall, 577 F.2d 301, 198 USPQ 344 (CCPA 1978).

The Cole et al. patent discloses a computer network 10 that includes a selection server 12 with a database 13, a content server 17, clients 14-16 and an Internet 20 to interconnect the servers and clients. (Col. 3, ll. 33-38.) A user at client computer 14 selects an icon to invoke an update manager program 32, which contacts a general manager program 31 of the selection server 12 to begin a session and supply the current level of update manager 32, scout 33, service application 34 and download routine 39 within client computer 14. (Col. 3, ll. 9-19.) The general manager 31 determines if this

BEST AVAILABLE COPY

13

client level information is the latest versions by comparing the same to the data in the selection server 12. (Col. 3, ll. 19-27.) If not the latest version, the server sends the FTP addressing information to the client computer for retrieving the latest versions from a content server, along with the FTP addressing information for a basic system information recognizer program 41 and installs the latest versions. (Col. 3, ll. 28-39.)

The recognizer program obtains basic system information about client 14 and sends this information to the selection server 12. (Col. 3, ll. 40-55.) As disclosed in Cole et al. this "basic system information comprises system model, pre-load software level, BIOS level, and information that is not likely to change often such as type of operating system." (Col. 3, ll. 47-50.) The update program 30 is initiated within server 12 to determine which code updates are consistent and inconsistent with the basic system information of the client. (Col. 3, ll. 56-63.) This determination is made by correlating meta data (descriptive information about the code updates such as Bios level, pre-load level, motherboard ID, etc. (col. 2, ll. 57-59), in database 13 to the basic system information obtained from client 14. (Col. 3, ll. 63-65.) Matches between the meta data and the basic system information indicate that the corresponding code updates are potentially appropriate for client computer 14. (Col. 3, l. 65 to col. 4, l. 1.)

For each code update that is consistent with the basic system information of client 14, the selection update program 30 sends to the client 14 the FTP addressing information of a corresponding recognizer program, such as recognizer programs 40 and 42, which are downloaded to the client 14 and executed thereon. (Col. 4, ll. 1-67.) The recognizer program 40 determines if the client 14 has the latest update by comparing the size of a

14

stored file to the size of such file stored in a database 13. (Col. 5, ll. 12-36.) Recognizer program 42 compares a version number of a program in the client 14 to the version number of such program of the code update, and if the program stored on the client computer has an older version number, then the recognizer program records that the client 14 needs a copy of the newer program. (Col. 5, ll. 37-55.)

These recognizer programs 40, 42 may assign a critical level to each code update based on the need of the client computer 14 for the update. (Col. 5, ll. 60-63.) After execution of these recognizer programs, the client computer 14 sends to the server 12 a list of code updates that are appropriate for the client 14 (step 120). Based on this information, the server determines the level of criticality of the code updates, and builds a selection form of a list of code updates consistent with the client computer (step 122) (Col. 6, ll. 1-8.) A user at the client computer 14 makes a selection and sends the selection to the server (steps 130, 131). (Col. 6, ll. 23-26.) In response, the server sends to the client computer 14 the FTP addressing information for the selected code updates (step 132), and the update codes are downloaded to the client computer 14. (Col. 6, ll. 26-30.)

The Invention

Applicant submits that independent claim 1, and claims 2-19 depend thereon, are directed to a method of updating computer software on a computer. The method includes downloading software update information through a network (e.g., the Internet) to the computer, and determining whether a software update is available from this downloaded software update information. A criticality check program is also downloaded to the computer through the network and executed on the computer. The criticality of the

software update is then evaluated using an output of the criticality check program and the software update information. Stored user preference information is accessed, and it is then determined if the software update should be automatically downloaded and installed using this user preference information and the evaluated criticality of the software update.

Cole et al. does not disclose or suggest a method of updating computer software on a computer by accessing stored user preference information, and then determining if a software update should be automatically downloaded and installed using the user preference information and an evaluated criticality of the software update. Rather, in Cole et al. it is a user at a client computer 14 that determines which code update to select from a list of updates and sends this selection (which the Examiner likens to applicant's stored user preference information) to the server. The server in Cole et al. then merely responds thereto by sending to the client computer the FTP addressing information for the selected code updates (step 132), and the update codes are downloaded to the client computer 14. (Col. 6, ll. 1-30.) That is, neither the server nor any other component of Cole et al. determines if a software update should be automatically downloaded and installed from stored user preference information, as is currently claimed. Again, in Cole et al. a user determines the selection and the server merely responds to such user-determined selection.

In the above office action the Examiner cites column 6, lines 46-55 of Cole et al. stating that the "download routine makes a determination whether any updates have been selected, based off the accessed stored user preference information, and automatically downloads the update, and then further, the server application routine automatically

installs the update. . . . As such, Cole does disclose storing the user's preference and using the same to automatically download and install an update as claimed." Firstly, applicant would like to clarify that this passage of Cole et al. occurs if there has been an interruption in the download of the code updates selected by the user, whereby the downloading can continue later where it left off. (Col. 6, ll. 35-42.) That is, they are based on the user-determined selection as discussed above. In the event of an interruption, "before the selected code updates are actually installed, the update manager 32 invokes the corresponding recognizer programs again to ensure that the code updates are still required If the code updates are still appropriate, the service application routine of client 14 installs the code updates in the client" (col. 6, ll. 41-48.)

Thus, in accordance with Cole et al., a user at the client computer makes a determination and selects a code update from a list of updates, whereby the selection information is sent to the server, and in response thereto, the server sends the information for the selected code update which is downloaded to the client computer. However, if an interruption occurs that causes the downloading process to stop, the selected code update may be later downloaded where left off by the update manager 32 invoking the recognizer programs to ensure that the code updates are still required, and if still appropriate, the service application routine of client 14 installs the user-selected code updates. The present invention includes limitations not taught or considered by Cole et al. since it is determined if a software update should be automatically downloaded and installed using user preference information and an evaluated criticality of the software update.

With respect to claim 4, applicant's invention includes further limitations not disclosed or contemplated by Cole et al. Claim 4 recites that the user preference information includes a user criticality threshold, such that the user criticality threshold and the evaluated criticality of the software update are compared in the step of determining if the software update should be automatically downloaded and installed. As disclosed in applicant's specification, a user sets the preferences as to what software updates should be automatically installed, "[f]or example, the user may set a preference that no automatic installations are allowed." (Specification, page 10, lines 4-11.) In so doing, the user may set a user criticality threshold in the user preference information for allowing or not allowing automatic installation of software updates, or set a separate criticality threshold for each software publisher or for each software product. (Specification, page 10, lines 13-27.) Independent claims 20, 25 and 26 also include this limitation that the user preference information includes a criticality threshold set by a user of the computer.

Unlike that of claims 4, 20, 25 and 26, Cole et al. discloses that the recognizer programs 40, 42 assign a critical level to each code update based on the need of the client computer 14 for the update, whereby the list of code updates are sent to the server, and the server determines the level of criticality of the code updates and builds a selection form of a list of update codes. (Col. 5, l. 60 to col. 6, l. 8.) It does not disclose a "user critical threshold" (i.e., user set preferences) whereby this user criticality threshold and the evaluated criticality of the software update are compared in the step of determining if the software update should be automatically downloaded and installed.

Dependent claim 12 recites that the method further includes the step of producing a notification if it is determined from the user preference information and the evaluated criticality of the software update that the software update should not be automatically downloaded and installed. Cole et al. does not recite this step of producing a notification if it is determined that the software update should not be automatically downloaded and installed. The Examiner cites column 4, lines 53-58, column 3, lines 56-67 and column 6, lines 1-58, however, it is submitted that nothing in these passages of Cole et al. disclose or suggest that a notification is produced if it is determined that the software update should not be automatically downloaded and installed. Rather, it is only disclosed that the selection update program 30 within server 12 determines which code updates are consistent and inconsistent with the basic system information of the client computer by correlating meta data in database 13 to the basic system information. (Col. 3, ll. 56-67.) Cole et al. discloses that based on this basic system information alone, the selection update program 30 is not sure which of the code updates are consistent with the basic system information. (Col. 4, ll. 53-58.) With respect to column 6, lines 1-58, nothing in this passage discloses or even suggests producing a notification.

With respect to claim 13, for the reasons discussed above, it is submitted that Cole et al. does not disclose determining if the software update should be automatically downloaded and installed by comparing a user criticality threshold included in the user preference information with a criticality rating included in the software update information. Cole et al. only discloses that the recognizer programs 40, 42 assign a critical level to each code update. (Col. 5, l. 60 to col. 6, l. 8.)

Independent claims 22-24 all recite that it is determined if the software update should be automatically downloaded and installed using accessed stored user preference information and evaluated criticality of the software update, such that these claims include limitations not disclosed or contemplated by Cole et al.

It is for these reasons that applicant submits that the present invention includes limitations not disclosed nor contemplated by Cole et al. such that Cole et al. does not anticipate nor render obvious the instant invention.

Section 103 Issues

The Examiner has also rejected claims 14-16, and 21 under 35 U.S.C. 103(a) as being unpatentable over Cole. Applicant disagrees.

As discussed above, Cole et al. does not anticipate nor render obvious the instant invention due to limitations in the invention which are not disclosed in Cole et al. In particular, Cole et al. does not disclose or suggest a method of updating software by determining if a software update should be automatically downloaded and installed using accessed stored user preference information and evaluated criticality of the software update, as is currently claimed. Cole et al. is limited to a user making a determination as to which code update to select, and this selection is sent to the server, which in response thereto sends the selected code updates to the client computer. (Col. 6, ll. 1-34.) As such, claims 14-16 (dependent on claim 1) and claim 21, all of which include this limitation of determining if a software update should be automatically downloaded and installed using accessed stored user preference information and evaluated criticality of the

software update, are also not anticipated by nor obvious over Cole et al., alone or in view of that which known in the art at the time of the invention.

Again, applicant submits that, as discussed in the "Description of Related Art" in applicant's application, automated download and install of updates are known methods in the art at which the present invention is aimed at overcoming. The present invention overcomes this problem by storing the user's preferences and using the computer, not the user, to make the choice between updates, guided by a "stored" user preference. On the contrary, Cole et al. teaches that a user at the client computer, not the computer, manually selects a desired update from a list of update codes.

The Examiner has also rejected claims 9-11 under 35 U.S.C. 103(a) as being unpatentable over Cole in view of Fawcett (U.S. Patent No. 5,845,077). Applicant disagrees.

Fawcett has been cited for the limitation that it teaches the step of producing a notification that the software update has been installed, as is recited in claim 9, and that it teaches the step of notifying the user of the availability of additional software updates, as is recited in claim 11. However, applicant submits that it does not overcome the deficiency of Cole et al. since it does not disclose or contemplate a method of updating software by determining if a software update should be automatically downloaded and installed using accessed stored user preference information and evaluated criticality of the software update, as is currently recited in claim 1, from which claims 9-11 depend. Rather, Fawcett is limited to a conventional prior art example at which the present invention is aimed at overcoming whereby a user must manually call an update service.

which inventories the user computer to determine what computer software may be out-of-date, and/or need maintenance updates, and if desired by the user, automatically downloads and installs computer software to the user computer. Abstract.

Claims 17 and 19 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Cole in view of Pedrizetti et al. (U.S. Patent No. 6,151,708). Applicant disagrees.

It is submitted that Pedrizetti et al. is limited to determining program update availability via set intersection over a sub-optical pathway by comparing a set of software programs on a client computer against a set of updates on a server computer to determine which updates are applicable and should be transferred from the server to the client. (Abstract.) The Examiner cites Pedrizetti et al. for the limitation that it "teaches the significance of a Dynamic Linked Library written by the vendor on the server computer associating a vendor to the updates to be downloaded by -that particular vendor (col. 6, lines 1-14). However, applicant continues to submit that neither Pedrizetti et al. nor Cole et al. disclose or suggest identifying one or more specific publishers in the claimed "software update query."

Pedrizetti et al. only discloses that a vendor may write a DLL that that coordinates any updates to the vendor's files and devices. (Col. 6, ll. 10-11.) Further, Pedrizetti et al. discloses sending this vendor specific DLL after the software update query has been sent by the client and after it has been determined that there is an update for software by that vendor. (See, col. 4, l. 43 to col. 6, l. 8 and Fig. 3.) This is not the same as a software

22

update query including software publisher information identifying a publisher of software to be updated, as is currently claimed.

Moreover, applicant submits that Pedrizetti et al. does not overcome the deficiencies of Cole et al. since it also does not disclose or contemplate a method of updating software by determining if a software update should be automatically downloaded and installed using accessed stored user preference information and evaluated criticality of the software update, as is claimed. It is for these reasons that applicant submits that claims 17 and 19 are neither anticipated by nor obvious over Cole et al. or Pedrizetti et al., alone or in combination.

Claim 18, which is dependent upon claim 17, and ultimately dependent upon claim 1, has been rejected under 35 U.S.C. 103(a) as being unpatentable over Cole and Pedrizetti, and further in view of Fawcett. However, for the reasons as discussed above, neither Cole et al., Pedrizetti et al. nor Fawcett, alone or in any proper combination thereof, disclose, contemplate or suggest a method of updating software by determining if a software update should be automatically downloaded and installed from accessed stored user preference information and evaluated criticality of the software update. As such, claim 18 is not obvious over Cole and Pedrizetti, and further in view of Fawcett.

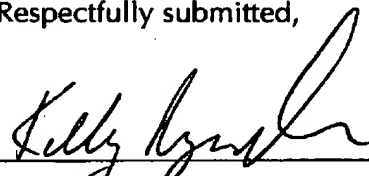
Applicant submits that the cited references of Cole et al., Pedrizetti et al. and Fawcett, alone or in any proper combination thereof, neither anticipate nor render obvious the instant invention.

It is respectfully submitted that the application has been brought into a condition where an allowance of the case is proper. Reconsideration of the pending claims and

23

issuance of a Notice of Allowance are respectfully solicited. Should the Examiner not find the claims to be allowable, Applicant's attorney respectfully requests that the Examiner call the undersigned to clarify any issue and/or to place the case in condition for allowance.

Respectfully submitted,



Kelly M. Reynolds
Reg. No. 47,898

DeLIO & PETERSON, LLC
121 Whitney Avenue
New Haven, CT 06510-1241
(203) 787-0595

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ BLACK BORDERS

☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☐ FADED TEXT OR DRAWING

☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☒ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.